DOCUMENT RESUME

ED 250 156 SE 045 129

AUTHOR	Lawlor, Joseph
TITLE	Artificial Intelligence and Expert Systems.
INSTITUTION	Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
SPONS AGENCY	National Inst. of Education (ED), Washington, DC.
REPORT NO	SWRL-TR-84
PUB DATE	15 Jan 84
CONTRACT	NEC-00-3-0064
NOTE	43p.
PUB TYPE	Reports - Descriptive (141) -- Information Analyses (070)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Artificial Intelligence; Chemistry; Cognitive Processes; Computer Science; *Computer Software; Engineering; Hazardous Materials; Medicine; Pattern Recognition; *Problem Solving; Robotics; Teaching Methods
IDENTIFIERS	*Expert Systems

ABSTRACT

Artificial intelligence (AI) is the field of scientific inquiry concerned with designing machine systems that can simulate human mental processes. The field draws upon theoretical constructs from a wide variety of disciplines, including mathematics, psychology, linguistics, neurophysiology, computer science, and electronic engineering. Some of the most promising developments to come out of recent AI research are "expert" systems or computer programs that simulate the problem-solving techniques of human experts in a particular domain. This paper reviews contemporary work in expert systems. It includes: (1) a brief history of AI research; (2) an overview of major lines of inquiry in the field, considering pattern recognition, robotics, and problem-solving; (3) a detailed discussion of the structure, design, and limitations of expert systems; and (4) a discussion of the role that expert systems might play in education. Two appendices are included. The first presents an example of interaction with an expert system; the second lists major systems that are currently in use or under development in the categories of bioengineering, chemistry, computer hardware, computer software, education, engineering, general purpose systems and artificial intelligence utilities, law, manufacturing and industry, mathematics, medicine, the military, and resource exploration. (JN)

SWRL EDUCATIONAL RESEARCH AND DEVELOPMENT

Artificial Intelligence and Expert Systems

2

TR 84 January 15, 1984

SWRL EDUCATIONAL RESEARCH AND DEVELOPMENT

TECHNICAL REPORT 84

January 15, 1984

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Joseph Lawlor

ABSTRACT

This paper reviews recent developments in artificial intelligence research, focusing on "expert" systems. Expert systems are computer programs that simulate the problem-solving skills of human experts in specific domains. The paper describes the structure, design, and limitations of such systems, and discusses their role in education.

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Joseph Lawlor

Artificial Intelligence (AI) is the field of scientific inquiry concerned with designing machine systems that can simulate human mental processes. The field draws upon theoretical constructs from a wide variety of disciplines, including mathematics, psychology, linguistics, neurophysiology, computer science, and electronic engineering. Some of the most promising developments to come out of recent AI research are "expert" systems--computer programs that simulate the problem-solving techniques of human experts in a particular domain. Reports in the popular media point out that these systems are the first AI products to demonstrate a potential for profit in the market place (Schrage, 1983; Shirley, 1983; Alexander, 1982b; Artificial Intelligence, 1982; Edelson, 1982; Yasaki, 1981).

This paper reviews contemporary work in expert systems. The paper first presents a brief history of AI research, followed by an overview of major lines of inquiry in the field. Next, it presents a detailed discussion of the design of expert systems. The paper concludes with a discussion of the role that expert systems might play in education. Appendix A presents an example of interaction with an expert system, and Appendix B lists major systems that are currently in use or under development.

History of Artificial Intelligence

McCorduck (1979) notes that several prominent AI authorities refer (only "half facetiously") to the classical, romantic, and modern ages

of AI research. Although these designations appear grandiose for a field that is no more than 30 years old,* the differences among the three periods are worth examining.

Classical Period

In the "classical" period, from the early 1950's to the late 1960's, the emphasis in AI was on a search for general principles of intelligence that could be applied to computer systems. Much of this early work was guided by the "cybernetic" model of intelligence, which attempted to draw an analogy between the electronic processes of computer circuits and the physiological processes of the human neuron. McCorduck (1979, p. 46) describes the cybernetic model:

> This modeling was not based on detailed biological knowledge of the natural cell, which we didn't have (nor for such purposes do we yet have). Instead, it seemed certain that the correspondence between the on-off behavior of the neuron and the on-off behavior of the electronic switch would be sufficient to allow significant modeling of neural systems, and than intelligent behavior.

However, the cybernetic model of intelligence proved to be inadequate for two major reasons. First, the incomplete description of neural behavior alluded to by McCorduck was more problematic than had originally been anticipated. AI researchers found it difficult to construct cybernetic systems that were anything more than trivial

*McCorduck (1979) points out that although serious work in AI could not begin until the advent of the digital computer in the 1950's, the "spiritual roots" of the field can be traced back as far as Greek mythology. The search for non-human intelligence is also reflected in the apocryphal stories surrounding medieval alchemists and mystics. More recently, the subject of machine intelligence was raised by the 19th century English mathematician, Charles Babbage, who proposed a design for an "analytical engine"--the forerunner of the modern computer.

examples of intelligence. Second, there appeared to be fundamental differences between the processing capabilities of the human brain and the digital computer. Computers--at least as we know them today--are essentially "serial" processors. That is, they execute instructions one at a time, in a step-by-step fashion. The brain, however, appears to be a "parallel" processor, capable of performing several activities simultaneously.

Consequently, the cybernetic model soon gave way to the "information-processing" model of intelligence. This model emphasizes the processes by which knowledge is symbolically stored, retrieved, and manipulated, regardless of how these processes are physically implemented in intelligent systems. Information processing has become the dominant paradigm for AI research over the past 25 years.

Romantic Period

From the late 1960's to the mid 1970's, the "romantic" age of AI research saw a growing emphasis on expanding the knowledge bases of intelligent systems. Important refinements to the information-processing model of intelligence were reflected in the field's concern for encoding and altering knowledge in machine-compatible form. Although this era produced encouraging developments, the field soon seemed to reach its limits. AI systems were capable of fairly sophisticated processing, but the systems did not seem to grow any "smarter."

Modern Period

In the "modern" age of AI, from the mid 1970's to the present, emphasis has shifted to the control aspects of AI systems. Control mechanisms became important as knowledge bases continued to expand due to more efficient coding techniques and improved computer hardware. Gradually, AI researchers adopted a three-part architecture for their systems: (1) a knowledge base, (2) a set of processes for transforming the knowledge base, and (3) a set of processes for controlling other processes in the system. This is the general architecture employed by modern expert systems.

Lines of Inquiry in Artificial Intelligence

In the United States, major AI projects were established at Stanford University, the Massachusetts Institute of Technology, Carnegie-Mellon University, and the Stanford Research Institute (now known as SRI International). From these and other projects, three major lines of inquiry have emerged: (1) pattern recognition, (2) robotics, and (3) problem-solving. Although it is useful to discuss these topics as distinct from one another, in reality they are all interrelated, particularly as they are exemplified in expert systems.

Pattern Recognition

AI work in pattern recognition includes two major subtopics, visual perception and natural-language processing. Researchers in the area of visual perception have attempted to design computer systems that can recognize alphabetical characters, as well as two- and three-dimensional

figures (Waltz, 1982). Results in this area have been encouraging, and operational systems have been developed for space agencies and the military (Alexander, 1982a).

The field of natural-language processing has achieved somewhat less impressive results. Early work in this area focused on designing automatic translation systems. Generally, these systems were disappointing, and they became favorite targets of AI critics (e.g., Taube, 1961).* Raphael (1976) notes that these early systems relied exclusively on a syntactic approach to translation. They simply looked up words in a dictionary and plugged them into the appropriate slots in sentences. However, it soon became apparent that such systems would have to "understand" the semantics of the text they were attempting to translate, greatly increasing the complexity of the task.

Nevertheless, AI researchers have achieved some important successes in natural-language processing. Winograd (1972) developed a program that could respond to natural-language commands and inquiries within its own limited domain, which consisted of a simulated mechanical arm that manipulated blocks. In other projects, intelligent query systems for large data bases were developed (Hendrix and Sacerdoti, 1981), and conversational programs that employed "scripts" of well known scenarios, such as going to a restaurant, showed impressive results (Abelson, 1981; Schank and Abelson, 1977).

*Other major works critical of AI research include Dreyfus (1972) and Weizenbaum (1976). Reviews of and rebuttals to this criticism can be found in McCorduck (1979) and Boden (1977).

Robotics

Early efforts in robotics research were concerned with designing general-purpose automatons that could interact with their environment to perform a variety of tasks. One of the first of these general-purpose robots was SHAKEY, developed by the Stanford Research Institute (Raphael, 1976). Although SHAKEY was capable of some remarkable feats (at least in his own narrowly defined environment), it soon became apparent that general-purpose robots were not cost-effective. Consequently, subsequent work in robotics has focused on special-purpose robots, primarily in industrial applications (e.g., Inoue, 1979).

Problem-Solving

The design of intelligent problem-solving systems has received a great deal of attention from the AI community. Within this broad area of inquiry, four major subtopics can be discerned. First are the game-playing systems that have been the traditional "workhorses" of research on problem-solving techniques. AI researchers found that games such as checkers and chess were ideal vehicles for exploring problem-solving because they offered environments that were reasonably limited and rule-bound, but that also posed interesting, complex problems.

Another facet of AI work in problem-solving involved the design of theorem-provers, programs that could construct logical proofs for mathematical principles (e.g., Green, 1969). An extension of this work with theorem-provers led to the third area of emphasis in problem-solving research--general problem-solving systems (e.g., Newell and Simon, 1963). Systems of this type were designed to employ logical principles to solve

a variety of problems, provided that the problems could be represented in the system's notation.

The final subtopic of the problem-solving line of inquiry focuses on expert systems. These are systems designed to solve problems in specific, well-bounded domains such as medical diagnosis or resource exploration. Such systems emulate the expertise of human professionals within the particular domain. Expert systems are described in detail in the next section of this paper.

AI researchers working in these four areas share a common concern with the general characteristics of problem-solving. The first major concern is the representation of the problem to be solved, whether that problem is one of responding to a particular move in chess or diagnosing a medical disorder. Raphael (1976, p. 52) notes that the following four conditions must be met before any problem can be subjected to formal analysis, by either human or artificial intelligence:

1. The problem is expressed in terminology clearly understood by the potential problem solver.
2. The form and notation for the problem's solution is agreed upon.
3. The relevant data upon which a solution may be based is identified.
4. Some measure of the validity or acceptability of proposed solutions is agreed upon.

Problem-solving systems in AI generally employ a "state-description" model of representation. That is, the problem is expressed as the initial state of the objects and processes within the domain of the problem (e.g., the current configuration of the chess pieces on the board). The solution is expressed as a "goal state" of the objects and

processes in the domain (e.g., an allowable configuration of chess pieces that gives one player the best advantage over the other).

Once the problem has been adequately represented, the problem-solving system must then conduct a search for the solution. This search can be illustrated by an inverted tree diagram (see Figure 1), in which the top node of the tree represents the initial state of the problem domain. The lines emanating from the nodes represent various actions or operators that can be applied in order to change the current state into the state represented by the lower node to which it is attached. The problem is solved when the system discovers a node that corresponds to the goal state.

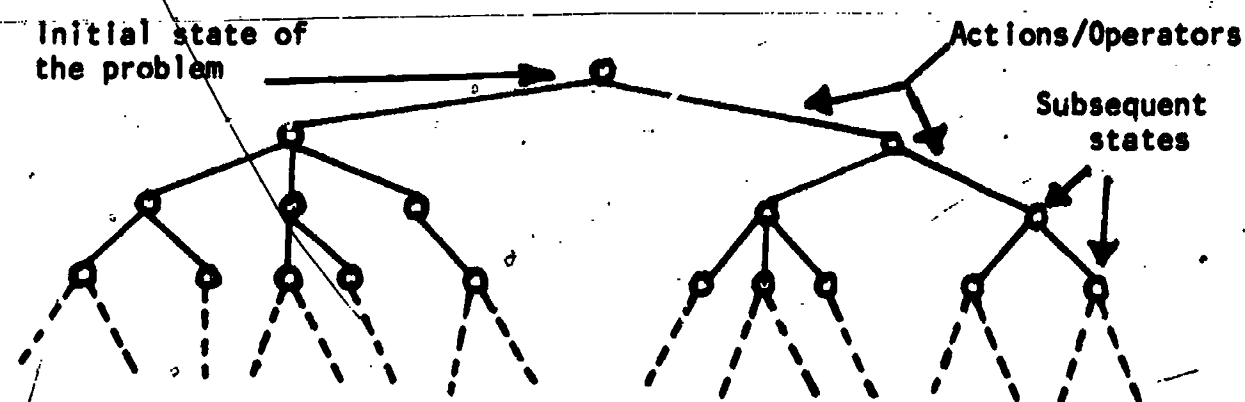

Figure 1. Partial Tree Diagram of a Problem-Solving Search.

Figure 1 also illustrates a critical obstruction in the design of problem-solving systems--the problem of the "combinatorial explosion." As is evident from the figure, the growth of the search space is exponential, as each node in the tree diagram generates one or more successors. If a problem-solving system were to explore every possible node in the tree, the problem would quickly become unmanageable. For

example, it has been estimated that there are approximately 10^{120} possible moves in a chess game. If the fastest modern computer were to calculate every conceivable move, the time required would surpass the expected life of the solar system (McCorduck, 1979, p. 157).

Consequently, problem-solving systems must employ techniques that limit the search space. AI researchers refer to this as "pruning" the search tree. Much of the effort in designing problem solvers has gone into developing heuristic techniques (Slagle, 1971; Lenat, 1982) that allow the programs to evaluate the probable efficiency of pursuing a particular branch of the search space. Thus, these systems can eliminate unproductive lines of inquiry and concentrate on branches that show promise of yielding a solution to the problem.

Expert Systems

Expert systems are programs that simulate the problem-solving techniques employed by human experts within a specific domain. These systems act as an "intelligent assistant," providing consultation and advice for a particular problem. For example, the first expert system to be developed, DENDRAL, was designed to identify chemical compounds, based upon data supplied by a mass spectrometer (Buchanan, Sutherland, and Feigenbaum, 1969). Initially, the program's performance was found to be equivalent to that of a doctoral candidate in chemistry. With its subsequent improvements, the program now surpasses human capabilities in this area (Feigenbaum and McCorduck, 1983, p. 62).

DENDRAL represented a significant departure from earlier work with problem-solving systems, in which the emphasis had been on representing general processes so that the systems could solve virtually any problem, regardless of the content area. However, the DENDRAL project suggested that content-specific knowledge was critical to problem-solving tasks.

Structure of Expert Systems*

Since the appearance of DENDRAL in the late 1960's, considerable effort has been expended in designing similar "knowledge-based" systems. (Appendix B of this paper lists more than 80 expert systems--both operational and experimental--that have been developed over the past 20 years.) These systems share a common architecture comprised of three subsystems: a knowledge base, an inference system, and an input/output system. Figure 2 illustrates how these subsystems are related in a typical expert system.

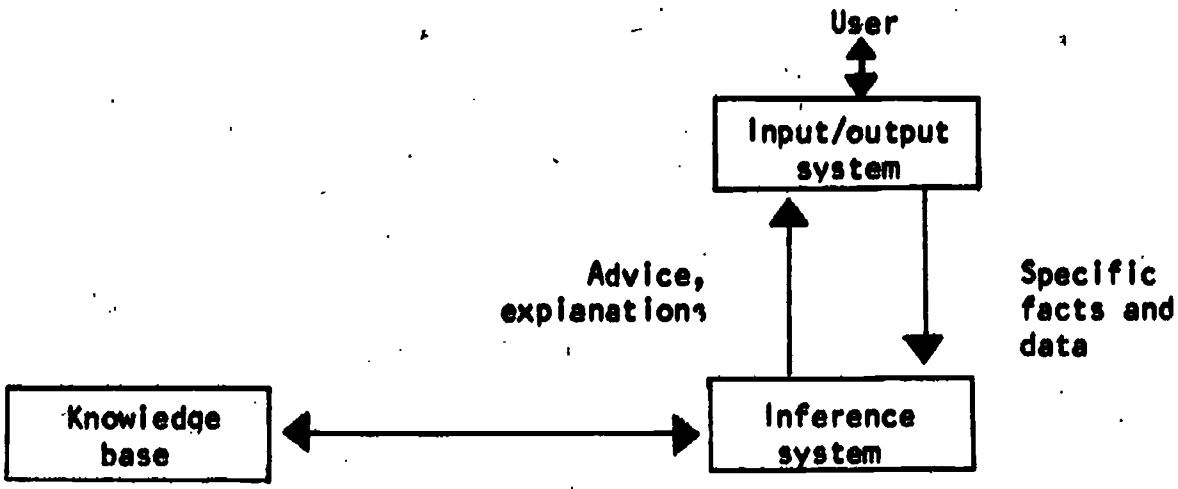

Figure 2. Basic Structure of an Expert System (based on Feigenbaum and McCorduck, 1983, p. 76).

*The discussion that follows is a summary of Feigenbaum and McCorduck (1983), pp. 76-84.

The knowledge base of an expert system consists of two types of knowledge. First is the factual knowledge of the particular domain--the commonly agreed upon information that appears in the textbooks and journals of the field. The second type is heuristic knowledge, the "rules of thumb" employed by human experts as they solve problems within the domain. Consequently, a knowledge base is significantly different from a data base, as Swaine (1983a) notes in the following analogy provided by Edward Feigenbaum, one of the pioneers in expert-systems research:

> A doctor, reading a patient's chart, is taking in data; but the doctor supplies the background knowledge to interpret that data. That knowledge is not acquired on the spot, but through years of medical school, internship, residency, specialization and practice in the field (p. 12; emphasis added).

The inference subsystem is the "reasoning engine" of the expert system. In most expert systems, the inference system consists of a series of "if-then" rules that guide the search for a solution to the problem (McCorduck, 1979, p. 287). A full-fledged expert system usually requires at least 500 such rules, and systems that employ more than a thousand rules are not uncommon (Duda and Gaschnig, 1981; Edelson, 1982).

The line of reasoning employed by the inference engine in an expert system must be explicit and easily comprehensible to the user. Such explicitness is necessary for debugging the program during its development, and for allowing the user to judge the practicality of the proposed solution. In some expert systems, the program also supplies the user with a "certainty measure," a numerical value that indicates the system's relative confidence in its conclusions (e.g., Duda and Gaschnig, 1981; Aikins, 1983).

The input/output system is the "user interface" to the expert system--the part of the program that gathers relevant data from the user and communicates the results of the problem-solving search. Usually, the input/output system also assists the user in specifying the problem. In addition, most expert systems employ some type of parsing system that allows the user to communicate in natural language (Swaine, 1983b; Feigenbaum, 1979). Appendix A illustrates how the input/output system of an expert system called ROSIE accomplishes these tasks.

It is important to note that expert systems are significantly different from the large data-base management systems in widespread use today. Such systems are capable of storing and recalling impressive amounts of information, yet they are essentially "unintelligent" in the sense that they possess no capabilities for altering their data or for performing the inferential reasoning that is the hallmark of expert systems. This is true even though some modern data-base systems employ natural-language query systems that allow the user to call up data using conversational English commands. Hendrix and Sacerdoti (1981) describe one such system called LADDER, which allows the user to access information about naval vessels. The program understands and responds to commands such as, "What U.S. merchant ships carrying vanadium ore are within 500 miles of the Kimlow?" (p. 314). Although the natural-language parser used in this program displays an impressive degree of intelligence, the actual retrieval of the data is accomplished through traditional data-search techniques.

Designing Expert Systems

The design of expert systems involves three major issues (Feigenbaum and McCorduck, 1983). First is the issue of knowledge representation--how to store, organize, control, and update the knowledge base of the expert system. This topic is a subject of much discussion in current AI literature (e.g., Waterman and Hayes-Roth, 1982; Aikins, 1983; Hawkins, 1983), and several approaches have proven useful. Nevertheless, imitating human methods of knowledge representation remains difficult, as Duda and Shortliffe (1983, p. 266) explain:

> It has frequently been noted that humans seem to exploit several different representations for the same phenomena. In particular, experts seem to employ rule-like associations to solve routine problems quickly, but can shift to using more reasoned arguments based on first principles when the need arises.

Continuing research in this area may lead to new techniques that will allow expert systems to replicate these multiple views of knowledge within the problem domain.

The second major design issue involves selecting the appropriate problem-solving strategies. Here again, several different approaches have been successful in particular domains (Duda and Shortliffe, 1983). However, one nagging question remains unanswered: What should an expert system do when it discovers that its strategies are inadequate for solving a particular problem? Feigenbaum (in Shea, 1983) describes one technique for dealing with this situation. If a system cannot answer a question, it can adopt a "graceful failure mode," in which the system responds with general information that is correct, but not necessarily useful. In this sense, the system does simulate a strategy often employed by human experts who find themselves in similar situations.

Clearly, though, more effective strategies must be developed for handling such cases.

The final issue in the design of expert systems has to do with knowledge acquisition, transferring expertise from human experts to machines, and enabling those machines to "learn" from their experiences. Knowledge acquisition is the most problematic of the three design issues, as Duda and Shortliffe (1983, p. 265) acknowledge:

> The very attempt to build a knowledge base often discloses gaps in our understanding of the subject domain and weaknesses in available representation techniques. Even when an adequate knowledge representation formalism has been developed, experts often have difficulty expressing their knowledge in that form.

The effective handling of these three design issues is obviously a complicated task. Consequently, the design of expert systems has evolved into a specialized branch of computer science called "knowledge engineering." The job of the knowledge engineer is to interview the human expert and translate the expert's factual and heuristic knowledge into a form that can be encoded by computer programmers. The knowledge engineer is also responsible for eliciting the problem-solving rules that make up the core of the inference system.*

The process of designing an expert system is a complex job that can take anywhere from 5 to 20 person-years to complete. Consequently, development of an expert system is an expensive proposition, with costs

---

*At the present time, there is a critical shortage of personnel in the field of knowledge engineering. According to one estimate, there are no more than 50 to 100 qualified knowledge engineers in the United States (Edelson, 1982). This shortage may hamper large-scale development of expert systems in the immediate future.

ranging from one to four million dollars (Edelson, 1982). However, steps have been taken to automate parts of the design process. One way is to provide the knowledge engineer with computerized tools for eliciting knowledge from the human expert. Figure 3 illustrates how these tools interact with the various parts of an expert system.

Another method for streamlining the design of expert systems is to build generalized rule-production systems that can be adapted to different problem domains. For example, an expert system called MYCIN was originally designed to diagnose blood infections and recommend treatment (Shortliffe, 1977). The inference subsystem of MYCIN was later adapted to a different knowledge base in a system called PUFF, which diagnosed lung disorders (Feigenbaum, 1979).

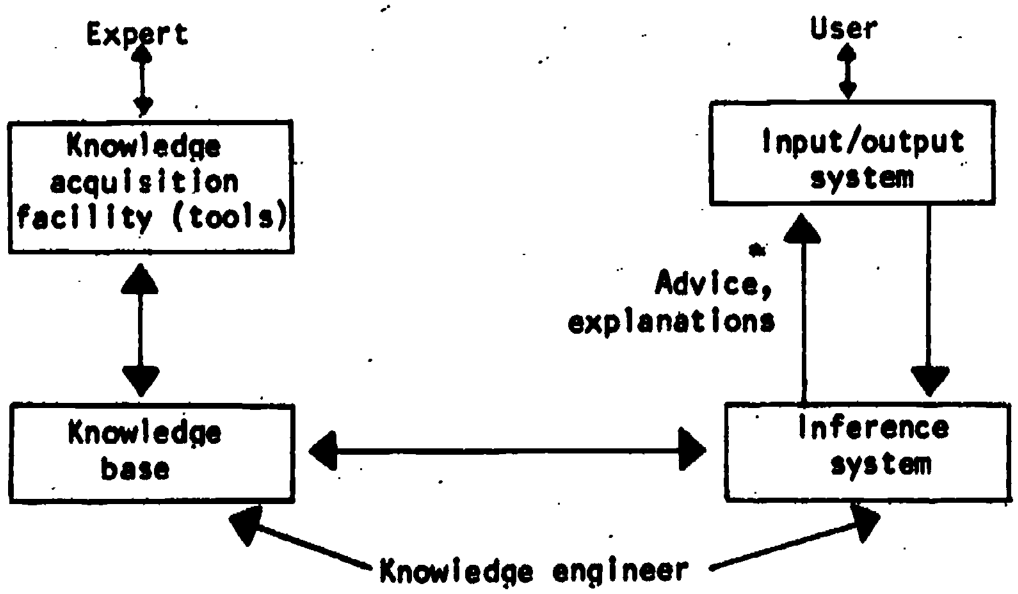

Figure 3. Designing an Expert System (from Feigenbaum and McCorduck, 1983, p. 76).

More recently, the Rand Corporation conducted an interesting experiment that compared the performance of eight rule-production systems

(Waterman and Hayes-Roth, 1982). Programming teams representing the eight systems were invited to design expert systems capable of responding to a simulated environmental crisis. The situation involved a chemical spill at the Oak Ridge National Laboratory in Tennessee, a complex of more than 200 buildings that store hazardous chemicals. The problem actually consisted of several subproblems, including identifying the spill material, warning people in the immediate area, containing the spill, finding the source of the material, and notifying the appropriate government agencies.

The situation also presented one additional problem: The system would have to be capable of handling changing conditions during the crisis. That is, the program would have to allow the user to enter new data supplied by field observers and respond to that data in a timely fashion. For example, if the program determined that the spill material was a noxious chemical, it would have to suspend all other processing and issue an immediate warning to workers in close proximity to the spill site.

The eight teams of programmers spent three days in intensive consulation with experts in the field and managed to produce prototype expert systems for dealing with the problem. (Appendix A illustrates how one of these systems approached the problem.) Because of the rigid time constraints imposed by the experimental design (normal development of an expert system can take months or years rather than days), the programming teams generally concentrated on only one or two of the subproblems. For example, the ROSIE system illustrated in Appendix A attempted to solve only two problems, identifying the spill material and determining the source of the spill.

The results of this experiment indicated that each of the production systems tested had its own relative strengths and weaknesses. For example, some systems handled the interactive aspects of the problem well, but had difficulty conducting the search for the spill source. This suggests that even generalized rule-production systems have to be tailored to the specifics of the problem domain.

Examples of Expert Systems

Expert systems have been designed to solve problems in a variety of fields, including medicine, engineering, computer hardware design, and resource exploration. The specific tasks performed by each of these systems varies with the nature of the particular problem. One way to survey the field of expert systems is to classify the systems by the type of problem that they attempt to solve. Stefik et al. (1982) provide a useful taxonomy that includes six general types of problem-solving tasks: interpretation, diagnosis, monitoring, prediction, planning, and design.

Interpretation. Interpretive expert systems are designed to analyze data and determine their significance. The DENDRAL program described above is an example of this type of system. Another kind of interpretation is performed by a system called HASP/SIAP, which analyzes data from sonar readings in order to identify and track naval vessels (Feigenbaum and McCorduck, 1983).

Diagnosis. Diagnostic systems perform fault-finding analyses. Many expert systems of this type are used in the medical field, for example, MYCIN and PUFF, which are described above. Diagnostic systems are also useful in the field of computer hardware maintenance. IBM, for example,

uses an expert system called DART to diagnose malfunctions in computer systems (Artificial Intelligence, 1982).

Monitoring. Monitoring systems interpret a continuous flow of data and sound an alarm when certain conditions are met. For example, an expert system called VM monitors intensive-care patients who are breathing with the assistance of a mechanical device (Fagan, 1980). If the system detects that the patient is having difficulties, it notifies medical attendants and recommends appropriate respiratory therapy. Another program, ACE, is used by Southwestern Bell to monitor maintenance of telephone lines (Schrage, 1983).

Prediction. Expert systems in this category are designed to make forecasts, based upon models of past experience. For example, the PROSPECTOR system analyzes geological data and predicts the likelihood of finding mineral deposits (Duda and Gaschnig, 1981). Recently, the system correctly predicted the presence of a large molybdenum deposit in the state of Washington (Shirley, 1983). Another predictive system, ROSS, is used by the Rand Corporation to conduct simulations of decision-making processes (Feigenbaum and McCorduck, 1983).

Planning. Expert systems that specialize in planning advise users on the optimum sequence of activities to achieve a particular goal. For example, MOLGEN is a system that aids in planning experiments in gene splicing (Stefik, 1981). In the field of engineering, a program called SACON provides advice on planning structural analyses (Feigenbaum and McCorduck, 1983).

Design. Expert systems in this category generate design specifications to meet specific requirements. Most systems of this type are used in the field of computer hardware design. For example, Xerox Corporation has developed an expert system called KBVLSI, which designs computer circuit boards (Feigenbaum and McCorduck, 1983). Digital Equipment Corporation uses a program called R1 to configure computer systems to match the needs of particular users (McDermott, 1982). Digital estimates that the R1 system has saved the company between seven and ten million dollars in the past three years (Schrage, 1983).

Expert Systems in Education

Some researchers (e.g., Papert, 1980; Gable and Page, 1980) have made a good case for the "promise" of AI in education, but that promise is still largely unrealized. Most uses of computers in elementary and secondary schools do not qualify as artificial intelligence by anybody's definition in this field of research. What schools have seen so far differs by an order of magnitude from the kind of applied AI research currently in use in science and industry.

To be sure, a wide variety of computer programs have been written for schools, but most of them have turned out to be fairly mundane applications known as "frame-oriented" systems. In a frame-oriented system, the computer simply "turns pages"--to be more precise, it skips through pages--in a programmed textbook. Such systems cannot be classified as "intelligent" because all of the instructional text, questions, and sequences for branching the learner from one frame to another are explicitly detailed by the program's author.

The "promise" in expert systems for education would be to provide "an infinitely patient, intelligent, and nonjudgmental tutor" (Feigenbaum and McCorduck, 1983, p. 89) that would reduce the tedious chore of specifying every single frame in an instructional program. Presumably, an expert system applied to education would be able to use its stored knowledge to generate appropriate questions and responses based on students' abilities and interests (Gable and Page, 1980; O'Shea, 1979).

Although expert systems in education sound too good to be true, several systems have been developed that do incorporate the technology of expert systems in educational applications. One of the first of these knowledge-based systems was SCHOLAR, developed by Bolt, Beranek, and Newman, Inc., in the late 1960's. The program is designed to teach factual knowledge about the geography of South America. SCHOLAR's knowledge base consists of a "semantic network," which is a set of "units," each of which is comprised of all the properties associated with that unit's label. Thus, for example, a unit labeled "Argentina" would have associated with it the properties of location, population, neighboring countries, and so on. These properties can themselves be pointers to other units within the network. For example, Argentina's "neighboring country" property would include a pointer to Uruguay, which is itself a separate unit. Consequently, the structure of SCHOLAR's knowledge base allows considerable complexity in the nesting of information (Gable and Page, 1980).

Using this knowledge base, SCHOLAR conducts a dialogue with the student, asking questions and responding to the student's questions as

well. What is significant about SCHOLAR is that these questions and responses are not "pre-programmed" as they are in frame-oriented systems. Thus, SCHOLAR represents a major departure from traditional approaches to computer-assisted instruction.*

Another knowledge-based teaching tool, SOPHIE, also developed by Bolt, Beranek, and Newman, guides students through a simulation that teaches the problem-solving skills involved in electronic troubleshooting (Brown, Burton, and Bell, 1975). SOPHIE presents the student with a schematic diagram of an electronic device and generates a fault in the device. The student's job is to identify and repair the malfunction by requesting various electrical measurements from SOPHIE and generating hypotheses about the location of the fault. The student may request any number of measurements in any order. If necessary, SOPHIE will, on request, generate several plausible hypotheses for the student to investigate. Consequently, SOPHIE does not rely on the relatively crude logic of traditional frame-oriented programs. Instead, SOPHIE employs a set of sophisticated inferencing procedures coupled with extensive knowledge about the behavior of electronic circuits. The program is also equipped with a sophisticated natural-language parser that can interpret misspellings, typographical errors, anaphoric references, and elliptical constructions. Figure 4 illustrates some of the capabilities of the SOPHIE system.

*Kearsley (1977-78) reports that a modified version of the SCHOLAR program has been developed to teach computer programming.

WHAT IS THE OUTPUT*

THE OUTPUT VOLTAGE IS 11.7 VOLTS.

WHAT IS IT IN A WORKING INSTRUMENT?

IN A WORKING CIRCUIT THE OUTPUT VOLTAGE IS 19.9 VOLTS.

WHAT IS THE VOLTAGE BETWEEN R8 AND Q1?

THE VOLTAGE AT N1 IS 47.6 VOLTS.

WHAT IS THE BSE EMITTER VOLTAGE OF THE VOLTAGE LIMITING TRANSITOR?

=BASE
=TRANSISTOR**
THE BASE EMITTER VOLTAGE OF Q5 IS -0.037 MILLIVOLTS.

WHAT ABOUT FOR THE CURRENT LIMITING TRANSISTOR?

THE BASE EMITTER VOLTAGE OF Q6 IS 0.349 VOLTS.

Figure 4. Sample Output from SOPHIE (from Brown, Burton, and Bell, 1975, p. 680)

*The student's input is shown in boldface.
**The program corrects two misspellings in the student's previous question.

Another instructional system that takes advantage of the technology of expert systems is GUIDON, which uses the knowledge base originally developed for MYCIN to teach diagnostic skills to medical students (Clancey, 1979). GUIDON includes a subsystem that emulates the expertise of a teacher. That is, the program constructs a model of the student's abilities and interests, based upon his or her responses to the program's questions. Then the system adjusts the presentation of instruction to match those abilities and interests, in much the same way that a human teacher would. The program presents the student with a sample patient case, and assists in formulating hypotheses about the cause of the

patient's infection. Duda and Gaschnig (1981) report that the GUIDON system has also been adapted to the knowledge base of the PUFF system, thereby providing a tutorial for pulmonary disease diagnosis.

Although expert systems seem to offer tremendous potential for improving computer-based instruction, several limitations must be overcome before expert systems come into widespread use in educational applications. One of the major limitations is their cost. As noted earlier, development and implementation of an expert system requires a great deal of effort and money. Thus, it is not surprising that most of the major systems now in use are designed for industrial and military environments, where research and development funding has traditionally been strong. Moreover, the cost-effectiveness of expert systems is readily evident in such settings: Systems tend to pay for themselves in a matter of a few years, as Digital Equipment Corporation has demonstrated with its R1 configuration system. In education, though, research funding is meager, and the cost-effectiveness of any system is much more difficult to demonstrate.

Closely related to cost is the problem of knowledge acquisition in building expert systems. Currently, the transfer of expertise from humans to machine systems is a painstaking art, requiring months or years of interviewing, programming, testing, and debugging. Although computer-based tools may aid in making this process more efficient, knowledge acquisition is likely to remain a bottleneck for some years to come.

Another limitation of expert systems is their applicability to particular problem domains. Expert systems are more successful at some tasks than at others, and it is difficult to determine their applicability to some domain in advance of actually attempting the application. Duda and Gaschnig (1981, p. 262) list the following prerequisites for a successful expert system:

1. There must be at least one human expert acknowledged to perform the task well.
2. The primary source of the expert's exceptional performance must be special knowledge, judgment, and experience.
3. The expert must be able to explain the special knowledge and experience and the methods used to apply them to particular problems.
4. The task must have a well-bounded domain of application.

Two final limitations of expert systems have to do with hardware constraints. First, most of the expert systems available today are written in specialized AI languages such as LISP, PLANNER, and PROLOG, which generally require large amounts of computer memory. Consequently, programs written in these languages usually demand at least a minicomputer environment in which to run. In educational settings, though, small microcomputers are becoming the standard. Nevertheless, we can expect this limitation to become less acute as microcomputer versions of the AI languages become available (Kewney, 1982), and as micros themselves become capable of addressing larger amounts of memory (Shea, 1983).

The second hardware limitation may prove more difficult to overcome. As was noted earlier in this paper, contemporary computers are serial

processors, which execute instructions one at a time. Although such prc ·sing is adequate for most applications that computers currently perform, AI programs are fast approaching the limits of serial machines. Many AI authorities feel that future developments in expert systems will require parallel processors--machines that can perform many calculations simultaneously and communicate the results of those calculations to other proc^^sors in the system (Feigenbaum and McCorduck, 1983; Alexander, 1982c).

Much of the major research and development work in parallel-processing machines has been undertaken by the Japanese in their "Fifth Generation" project. This project, which involves a consortium of government agencies and private companies, has received a government commitment of $450 million over the next ten years to develop hardware and software for advanced AI applications. With additional matching contributions from private industry, the budget of the Fifth Generation project may eventually exceed $850 million (Feigenbaum and McCorduck, 1983, p. 106). The project is still in its infancy, and it is not clear that any major breakthroughs are imminent. In fact, many American experts are skeptical. Uttal, for example, quotes one American executive as saying that the Fifth Generation "has more of a public relations flair to it than reality" (1982, p. 82).

Nevertheless, it may well be that the Japanese project will produce important breakthroughs in hardware and software systems. In addition, the Fifth Generation Project appears to be stimulating similar development efforts in this country and in Europe. This renewed interest in AI research is likely to have profound effects on the future of expert systems in all domains, including education.

REFERENCES

Abelson, R. P. Psychological status of the script concept. American Psychologist, 1981, 36, 715-729.

Aikins, J. S. Prototypical knowledge for expert systems. Artificial Intelligence, 1983, 20, 163-210.

Alexander, T. Teaching computers the art of reason. Fortune, May 17, 1982, pp. 82-92. (a)

Alexander, T. Practical uses for a "useless" science. Fortune, May 31, 1982, pp. 138-145. (b)

Alexander, T. Computers on the road to self-improvement. Fortune, June 14, 1982, pp. 148-160. (c)

Artificial intelligence: The second computer age begins. Business Week, March 8, 1982, pp. 66-75.

Boden, M. A. Artificial intelligence and natural man. New York: Basic Books, 1977.

Brown, J. S., Burton, R. R., & Bell, A. G. SOPHIE: A step toward creating a reactive learning environment. International Journal of Man-Machine Studies, 1975, 7, 675-696.

Buchanan, B., Sutherland, G., & Feigenbaum, E. A. HEURISTIC DENDRAL: A program for generating hypotheses in organic chemistry. In B. Meltzer & D. Michie (Eds.), Machine intelligence 4. New York: American Elsevier, 1969.

Clancey, W. J. Tutoring rules for guiding a case method dialogue. International Journal of Man-Machine Studies, 1979, 11, 25-49.

Dreyfus, H. L. What computers can't do. New York: Harper & Row, 1972.

Duda, R. O., & Gaschnig, J. G. Knowledge-based expert systems come of age. Byte, September 1981, pp. 238-281.

Duda, R. O., & Shortliffe, E. H. Expert systems research. Science, 1983, 220, 261-268.

Edelson, E. Expert systems: Computers that think like people. Popular Science, September 1982, pp. 48-58.

Fagan, L. M. VM: Representing time-dependent relations in a medical setting (Doctoral dissertation, Stanford University, 1980). Dissertation Abstracts International, 1980, 41, 1827-8. (University Microfilms No. 8024651)

Feigenbaum, E. A. Themes and case studies of knowledge engineering. In D. Michie (Ed.), Expert systems in the microelectronic age. Edinburgh, Scotland: University of Edinburgh Press, 1979.

Feigenbaum, E. A., & McCorduck, P. The fifth generation: Japan's computer challenge to the world. Reading, MA: Addison-Wesley, 1983.

Gable, A., & Page, C. V. The use of artificial intelligence techniques in computer-assisted instruction: An overview. International Journal of Man-Machine Studies, 1980, 12, 259-282.

Green, C. Theorem-proving by resolution as a basis for question-answering systems. In B. Meltzer & D. Michie (Eds.), Machine Intelligence 4. New York: American Elsevier, 1969.

Hawkins, D. An analysis of expert thinking. International Journal of Man-Machine Studies, 1983, 18, 1-47.

Hendrix, G. G., & Sacerdoti, E. D. Natural language processing: The field in perspective. Byte, September 1981, pp. 304-352.

Inoue, H. Force feedback in precise assembly tasks. In P. H. Winston and R. H. Brown (Eds.), Artificial intelligence: An MIT perspective (Vol. 2). Cambridge, MA: MIT Press, 1979.

Kearsley, G. P. The relevance of AI research to CAI. Journal of Educational Technology Systems. 1977-78, 6, 229-251.

Kewney, G. The other side. Creative Computing, November 1982, pp. 282-283.

Lenat, D. B. The nature of heuristics. Artificial Intelligence, 1982, 19, 189-249.

McCorduck, P. Machines who think. San Francisco: W. H. Freeman, 1979.

McDermott, J. R1: A rule-based configurer of computer systems. Artificial Intelligence, 1982, 19, 39-88.

Newell, A., & Simon, H. A. GPS, a program that simulates human thought. In E. A. Feigenbaum and J. A. Feldman (Eds.), Computers and thought. New York: McGraw-Hill, 1963.

O'Shea, T. Rule-based computer tutors. In D. Michie (Ed.), Expert systems in the microelectronic age. Edinburgh, Scotland: University of Edinburgh Press, 1979.

Papert, S. Mindstorms. New York: Basic Books, 1980.

Raphael, B. The thinking computer. San Francisco: W. H. Freeman, 1976.

Schank, R. C., & Abelson, R. P. Scripts, plans, goals and understanding. Hillsdale, NJ: Lawrence Erlbaum, 1977.

Schrage, M. Profits in artificial intelligence. Washington Post, August 28, 1983, pp. G1; G3.

Shea, T. Future data bases will be smarter rather than bigger. Infoworld, May 23, 1983, pp. 34-35.

Shirley, S. How expert systems decide. Management Today, February 1983, pp. 33-38.

Shortliffe, E. H. MYCIN: A knowledge-based computer program applied to infectious diseases. In Proceedings of the first annual symposium on computer application in medical care. New York: IEEE, 1977.

Slagle, J. R. Artificial intelligence: The heuristic programming approach. New York: McGraw-Hill, 1971.

Stefik, M. Planning with constraints. Artificial Intelligence, 1981, 16, 111-140.

Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F., & Sacerdoti, E. The organization of expert systems, a tutorial. Artificial Intelligence, 1982, 18, 135-173.

Swaine, M. Computer-AI link revolutionizes symbolic sciences. Infoworld, February 14, 1983, pp. 12; 14. (a)

Swaine, M. Natural-language interfaces for data-base systems. Infoworld, February 28, 1983, pp. 21-22. (b)

Taube, M. Computers and common sense. New York: Columbia University Press, 1961.

Uttal, B. Here comes Computer Inc. Fortune, October 4, 1982, pp. 82-90.

Waltz, D. L. Artificial intelligence. Scientific American, 1982, 247(4), 118-133.

Waterman, D. A., & Hayes-Roth, F. An investigation of tools for building expert systems (Report No. R-2818-NSF). Santa Monica, CA: Rand Corporation, 1982.

Weizenbaum, J. Computer power and human reason. San Francisco: W. H. Freeman, 1976.

Winograd, T. Understanding natural language. New York: Academic Press, 1972.

Yasaki, E. K. AI: More than a science. Datamation, September 1981, pp. 63-64.

APPENDIX A

INTERACTION WITH THE EXPERT SYSTEM ROSIE*

Go investigate**

Ready for report: **The spill is detected at WOC-6***.**
The solubility of the spill does appear low.
The pH of the spill does approximate 8.
End of report.

Report received.

'SPILL #1 is detected at WOC-6' has been recorded.
'SOLUBILITY #1 does appear LOW' has been recorded.
'PH #1 does approximate 8' has been recorded.

Now attempting to determine the material type.
'TYPE-OF-MATERIAL #1 is an oil.'

Now attempting to determine the spill-material.
'SODIUM HYDROXIDE is not a possible-material of SPILL #1.'
'POTASSIUM HYDROXIDE is not a possible-material of SPILL #1.'
'SULFURIC ACID is not a possible-material of SPILL #1.'
'HYDROCHLORIC ACID is not a possible-material of SPILL #1.'
''ACETIC ACID is not a possible-material of SPILL #1.'

Now attempting to determine the spill volume.

Now trying to determine the source of the spill.
Locations eliminated (wrong material):

(A list of locations follows.)

What does the spill smell like?
Please characterize the spill's appearance: a colorless film, a silvery film, a lightly colored film, a brightly colored film, a film with colors turning dull or a film with very dark colors.
What is the area of the spill (in square meters)?

---

*This material is taken from Waterman and Hayes-Roth (1982, pp. 66-71). ROSIE is a registered trademark of the Rand Corporation.

**The user's input is shown in boldface.

***These alphanumeric references refer to mapped locations at the Oak Ridge National Laboratory.

Please check M6-15 for signs of the spill.

Ready for report: The spill is detected at m6-15.
The spill does have no odor.
End of report.

Report received.

'SPILL #1 is detected at M6-15' has been recorded.
'SPILL #1 does have NO ODOR' has been recorded.

Now attempting to determine the spill-material.
'GASOLINE is not a possible-material of SPILL #1.'
'DIESEL OIL is not a possible-material of SPILL #1.'

Now attempting to determine the spill volume.

Now trying to determine the source of the spill.
Locations eliminated (below a dirty inspection site):

(A list of locations follows.)

Locations eliminated (wrong material):

(A list of locations follows.)

Please characterize the spill's appearance: a colorless film, a silvery film, a lightly colored film, a brightly colored film, a film with colors turning dull or a film with very dark colors.
What is the area of the spill (in square meters)?
Please check M6-29 for signs of the spill.

Ready for report: The spill is detected at m6-29.
The spill does form (a film with colors turning dull).
The spill does cover 78000 m**2.
End of report.

Report received.

'SPILL #1 is detected at M6-29' has been recorded.
'SPILL #1 does form FILM #1' has been recorded.
'SPILL #1 does cover 78000 M**2' has been recorded.

Now attempting to determine the spill-material.

Now attempting to determine the spill volume.
'SPILL #1 does involve 20.592 GALLONS OF MATERIAL.'

Now trying to determine the source of the spill.
Locations eliminated (below a dirty inspection site):

(A list of locations follows.)

Locations eliminated (too little volume):

(A list of locations follows.)

Please check M6-34 for signs of the spill.

Ready for report: The spill is not detected at m6-34. End of report.

Report received.

'SPILL #1 is not detected at M6-34' has been recorded.

Now attempting to determine the spill-material.

Now trying to determine the source of the spill.
Locations eliminated (below a dirty inspection site):

(A list of locations follows.)

Please check M6-40 for signs of the spill.

Ready for report: The spill is not detected at m6-40. End of report.

Report received.

'SPILL #1 is not detected at M6-40' has been recorded.

Now attempting to determine the spill-material.

Now trying to determine the source of the spill.
Locations eliminated (above a clean inspection site):

(A list of locations follows.)

Please check M6-35 for signs of the spill.

Ready for report: The spill is not at m6-35.

Improper report form.
Please try again.

The spill is not detected at m6-35. End of report.

Report received.

'SPILL #1 is not detected at M6-35' has been recorded.

Now attempting to determine the spill-material.

Now trying to determine the source of the spill.
Locations eliminated (above a clean inspection site):

(A list of locations follows.)

Please check M6-46 for signs of the spill.

Ready for report: The spill is detected at m6-46.
End of report.

Report received.

'SPILL #1 is detected at M6-46' has been recorded.

Now attempting to determine the spill-material.

Now trying to determine the source of the spill.
Locations eliminated (above a clean inspection site):

(A list of locations follows.)

Please check S6-37 for signs of the spill.

Ready for report: The spill is detected at s6-37.
End of report.

Report received.

'SPILL #1 is detected at S6-37' has been recorded.

Now attempting to determine the spill-material.

Now trying to determine the source of the spill.
Locations eliminated (below a dirty inspection site):

M6-46

The spill does originate at S6-37.
The material of the spill is hydraulic oil.
The source is located in BUILDING 3024.

APPENDIX B

SELECTED EXPERT SYSTEMS*

Domain	Program/Description	Institution
Bioengineering	GEL: determines the appropriate sequence of nucleic acids	IntelliGenetics, Ltd.
	GENESIS: plans gene-splicing experiments	IntelliGenetics, Ltd.
	MOLGEN: plans experiments in genetic engineering	Stanford University
	PEP: analyzes protein sequences	IntelliGenetics, Ltd.
Chemistry	DENDRAL: determines molecular structure from mass spectrometer data	Stanford University
	SECS: plans procedures for organic chemical synthesis	University of California, Santa Cruz
Computer Hardware	DART: diagnoses computer system faults	Stanford University and IBM
	DESIGN AUTOMATION ASSISTANT: aids in design of computer circuits	Bell Laboratories
	ISA: determines the appropriate delivery date for computer systems	Digital Equipment Corporation
	KBVLSI: designs computer circuits	Xerox Corporation and Stanford University
	MAPLE: designs micro-processor hardware	University of Reading, England

*Much of the information in this appendix is taken from Feigenbaum and McCorduck (1983, pp. 244-250). Their original list has been supplemented by the author.

Domain	Program/Description	Institution
	R1 (also known as XCON): configures computer systems	Carnegie-Mellon University and Digital Equipment Corporation
	SPEAR: analyzes computer error logs	Digital Equipment Corporation
	XSEL: assists sales-persons in selecting computer systems	Digital Equipment Corporation
	QUESTWARE: provides advice on selecting small computer systems	Dynaquest Consulting Groups
Computer Software	PROGRAMMER'S APPRENTICE: assists in constructing and debugging programs	Massachusetts Institute of Technology
	PROJCON: analyzes problems in software development projects	Georgia Institute of Technology
	PSI: composes computer programs based on an English description of the required task	Kestrel Institute
	VISUAL DESIGN CONSULTANT: aids pro-grammers in formatting menu displays for software applications	Virginia Polytechnic Institute
Education	BLOCKS: teaches problem-solving skills through the use of attribute blocks and Venn diagrams	Cleveland State University
	GUIDON: teaches medical diagnostic skills	Stanford University
	LMS: diagnoses student errors in solving arith-metic problems and algebraic equations	Leeds University, England

Domain	Program/Description	Institution
	SCHOLAR: teaches South American geography	Bolt, Beranek, and Newman, Inc.
	SOPHIE: general-purpose tutorial system; has been used to teach electronic trouble-shooting	Bolt, Beranek, and Newman, Inc.
Engineering	ACE: monitors and analyzes telephone line maintenance	Bell Laboratories
	PEACE: analyzes electronic circuits	Onera-Cert, Toulouse, France
	SACON: assists structural engineers in planning analyses	Stanford University
	SPERIL: assesses structural damage for civil engineers	University of Tokyo
General-Purpose Systems and Artificial Intelligence Utilities	AGE: assists in developing expert systems	Stanford University
	AL/X: encodes scientific knowledge into an intelligent data base	Intelligent Terminals, Ltd.
	ASK: general-purpose knowledge-based system	California Institute of Technology, Pasadena
	CAKES: general-purpose system for designing other expert systems	Teknowledge, Inc.
	EMYCIN: basic inference system used as the architecture for other expert systems (e.g., MYCIN, PUFF)	Stanford University

Domain	Program/Description	Institution
	EURISKO: general-purpose inference system capable of generating its own heuristics; has been used in computer circuit design	Stanford University
	EXPERT: basic inference system used in the petroleum and medical fields; has been used to diagnose thyroid diseases	Rutgers University
	HEARSAY: tool for building expert systems	Carnegie-Mellon University
	ICLX: general-purpose expert system	Systems Strategy Centre, ICI, Stevenage England
	KAS: creates rule networks for expert systems	SRI International
	KEPE: general-purpose knowledge representation system	IntelliGenetics, Ltd.
	KM-1: general-purpose knowledge management system	System Development Corporation
	LOOPS: general-purpose knowledge representation system used in KBVLSI	Xerox Corpporation
	MERLIN: conducts and analyzes experiments in artificial intelligence	Carnegie-Mellon University
	MLSE: assists in designing knowledge-based systems	University of Tokyo
	MRS: general-purpose knowledge representation system	Stanford University
	OPS: basic inference system used in other expert systems	Carnegie-Mellon University

Domain	Program/Description	Institution
	RABBIT: assists in formulating queries for a data base	Xerox Corporation
	ROSIE: general-purpose inference system	Rand Corporation
	ROSS: conducts decision-making simulations	Rand Corporation
	SAGE: general-purpose inference system	SRI International
	SPEX: analyzes human speech through the use of spectrograms	Verbex Corporation
	TEIRESIAS: assists in building knowledge bases for expert systems	Stanford University
	UNITS: general-purpose knowledge representation system	Stanford University
Law	ELI: consultation system for welfare rights workers	Open University, Milton Keynes, England
	LDS: assists lawyers and claims adjusters	Rand Corporation
Manufacturing and Industry	CALLISTO: schedules and monitors large projects	Carnegie-Mellon University
	FRUMP: summarizes wire-service news stories	United Press International
	ISIS: conducts job scheduling	Carnegie-Mellon University
	KBCS: controls manufacturing in electronics plants	Mitsubishi Electric Corporation, Japan
	KS-300: provides industrial diagnosis and advice	Teknowledge, Inc.
	TAXADVISOR: provides advice for federal tax planning	University of Nebraska

Domain	Program/Description	Institution
	TAXMAN: assists in corporate tax planning	Rutgers University
	TDUS: assists mechanics in repairing electro-mechanical equipment	SRI International
Mathematics	MACYSMA: mathematician's assistant for solving algebraic expressions	Massachusetts Institute of Technology
Medicine	ABEL: diagnoses medical disorders	Massachusetts Institute of Technology
	CADUCEUS: performs diagnoses in internal medicine	University of Pittsburgh
	CASNET: recommends treatments for glaucoma	Rutgers University
	CENTAUR: diagnoses lung disorders	Xerox Corporation
	MECS-AI: provides diagnostic assistance	University of Tokyo
	MICRORHEUM: assists in diagnosis and treatment of rheumatism	University of Missouri
	MYCIN: diagnoses blood infections	Stanford University
	ONCOCIN: manages cancer chemotherapy	Stanford University
	PUFF: diagnoses lung disorders	Stanford University
	RECONSIDER: provides medical diagnoses	University of California Medical Center, San Francisco
	VM: monitors patients in intensive care; recommends respiratory therapy	Stanford University

Domain	Program/Description	Institution
Military	ACTS: teaches decision-making skills in military environments	Perceptronics, Inc.
	AIRPLAN: manages air traffic around aircraft carriers	Carnegie-Mellon University
	AIS: selects and presents messages in tactical military settings	Perceptronics, Inc.
	GRACIE: translates and correct telegraphic messages	Hughes Aircraft Company
	HASP/SIAP: uses sonar signals to track ships	Systems Control Technology, Inc. and Stanford University
	KNOBS: assists in planning Air Force missions	Mitre Corporation
	TATR: provides advice on air combat tactics	Rand Corporation
Resource Exploration	DIPMETER ADVISOR: analyzes information from oil well logs	Schlumberger-Doll Research
	DRILLING ADVISOR: diagnoses oil well drilling problems	Teknowledge, Inc., and Elf-Aquitaine
	HYDRO: assists in solving water resource problems	SRI International
	LITHO: performs geological analyses	Schlumberger-Doll Research
	PROSPECTOR: evaluates sites for potential mineral deposits	SRI International
	WAVES: provides advice on seismic data in oil exploration	Teknowledge, Inc.